

## Lab- 10

### CSET340- Advanced Computer Vision and Video analytics

#### Task-1:- Image Denoising, Extract images from video, Process images of a video

**Image Denoising:-** Take three different types of image denoising techniques namely **Median filter**, **Wavelet denoising filter** and **Noise2Void** to denoise same set of images and compare their Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Mean Squared Error (MSE).

Links:- <https://www.youtube.com/watch?v=71wqPyapFGU>

[https://scikit-image.org/docs/0.25.x/auto\\_examples/filters/plot\\_denoise.html](https://scikit-image.org/docs/0.25.x/auto_examples/filters/plot_denoise.html)

<https://www.kaggle.com/code/salmankhaliq22/opencv-skimage-denoising-techniques>

**Extract images from video:-** Take any video (preferably .mp4) and extract the frames of the video using OpenCV library cv2.VideoCapture, cv2.read( ) and cv2.imwrite( ) functions.

**Process images of a video:-** Take any video (preferably .mp4) and perform the fundamental image processing **adaptive thresholding**, **Gaussian smoothing**, **canny edge detection** and **bitwise-not** operations separately on the video frame by frame. And, finally recreate the new videos from the transformed frames. Also, build a collage from the video frames.

<https://www.geeksforgeeks.org/extract-images-from-video-in-python/?ref=lbp>

<https://www.geeksforgeeks.org/python-process-images-of-a-video-using-opencv/?ref=lbp>

<https://www.kaggle.com/code/arielfelices/video-analysis-using-opencv-and-numpy>

#### Task-2:- Video classification using CNN+LSTM or 3D CNN model.

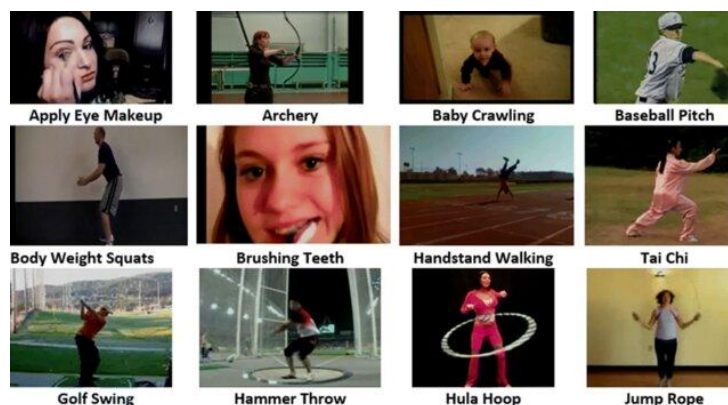
To build a video classification pipeline using a subset of the UCF101 dataset.

**UCF101** is a dataset of 13,320 video clips across **101** human action categories. Each video is a .avi file showing a single action (e.g., basketball, biking, playing guitar).

Dataset URL: <https://www.crcv.ucf.edu/data/UCF101.php>

<https://www.kaggle.com/datasets/pevogam/ucf101>

<https://www.kaggle.com/datasets/matthewjansen/ucf101-action-recognition>



Use a subset of **5 action classes**, each with **10 videos only** with any 5 from the following classes: Basketball, Biking, PlayingGuitar, Typing, JumpRope, YoYo, HorseRiding, Archery.

To select a subset of the datasets either **manually download** the file corresponding to the classes mentioned or Automated Filtering by creating a new subset directory using script:

```
import os

import shutil

import random

SOURCE_DIR = '/content/UCF-101' # Original full dataset path

DEST_DIR = '/content/UCF101_subset'

SELECTED_CLASSES = ['Basketball', 'Biking', 'PlayingGuitar', 'Typing', 'JumpRope']

VIDEOS_PER_CLASS = 10

os.makedirs(DEST_DIR, exist_ok=True)

for cls in SELECTED_CLASSES:

    class_path = os.path.join(SOURCE_DIR, cls)

    dest_class_path = os.path.join(DEST_DIR, cls)

    os.makedirs(dest_class_path, exist_ok=True)

    selected = random.sample(os.listdir(class_path), VIDEOS_PER_CLASS)

    for video in selected:

        shutil.copy(os.path.join(class_path, video), dest_class_path)
```

### Preprocess:-

- ❖ Extract frames from each video (e.g., every 5th frame).
- ❖ Resize frames to 112×112 or 224×224.
- ❖ Convert frame sequences into a fixed length (e.g., 16 frames per video).
- ❖ Label encode class names.
- ❖ Split the dataset into training and testing (80/20).

Apply models either of **CNN-LSTM** or **3D- CNN** and Evaluate the model using: Accuracy, Confusion matrix, Per-class Precision, Recall and F1-score.

Links for help:-

<https://www.analyticsvidhya.com/blog/2019/09/step-by-step-deep-learning-tutorial-video-classification-python/#h-exploring-the-video-classification-dataset>

<https://www.v7labs.com/blog/video-classification-guide>

<https://github.com/HHTseng/video-classification>

<https://github.com/Easwar-esh/Video-classification-using-cnn-lstm>

[https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/video/video\\_classification.ipynb](https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/video/video_classification.ipynb)

**Note:-** Submit the notebook file on LMS and Git link.

Marks will be deduced for late submission.